A Note on Some Tree Similarity Measures

Karel Culik II
Derick Wood

CS-82-30

September, 1982

A NOTE ON SOME TREE SIMILARITY MEASURES[1]

by

Karel Culik II[2] and Derick Wood[2]

(1) Work carried out under Natural Sciences and Engineering Council of Canada Grant Nos. A-7403 and A-7700.

(2) Department of Computer Science
University of Waterloo
Waterloo, Ontario
N2L 3G1 Canada

1. Introduction

Given two structures of the same type, one standard question is: how close are these two structures to each other? One example is the string-to-string correction problem, see [H], [S], and [WF] for example, a second is syntax-error repairing in parsers, see [BP], and [L], and a third is the similarity of two dendrograms [D], [MGB] and [WS].

It is this third example we are concerned with in the present note. We consider labelled and unlabelled trees and search trees of the same size n. We show that two trees of the same type are O(n) and O(nlogn) distance apart, for unlabelled and labelled trees respectively. The basis for the distance measure is the interchange or rotation tree transformation.

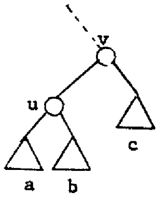
2. The Results

We define an (unrooted) binary tree (or dendrogram) to be a connected graph with no cycles, where each node is either unary or ternary. A unary node is called a leaf, external or terminal node, while a ternary node is called an internal node. If a binary tree has $n > 3$ leaves, then it has $n - 2$ internal nodes and $n - 3$ edges connecting the internal nodes.

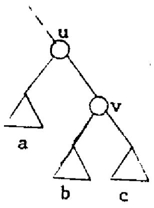
Similarily we define a rooted, oriented and ordered binary tree, usually called a binary search tree to be a connected digraph, with a designated root node. Each node apart from the root has in-degree one and out-degree either 0 or 2. The former are leaves and the latter are internal nodes.

A well known tree transformation in binary search trees is that of rotation or promotion, namely:

given



a clockwise rotation about v yields:



A counterclockwise rotation about u in the second diagram yields the tree in the first diagram. The reasons for the importance of this rotation are two-fold. First, if the original tree is a valid search tree, that is its nodes are also canonically labelled with "keys" from some universe, then the resulting tree is also a valid search tree. Second, it affects the "balance" of the rotation node, and this is exploited in various balanced binary search tree schemes, see [K, pp. 451 ff] for example.

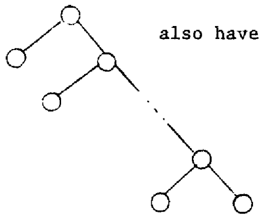
Given this transformation, which we denote by ρ, and two binary search trees S and T each with n nodes, we can define the <u>rotation distance of S and T</u>, denoted by <u>rdist(S,T)</u>, as: the minimum number of applications of ρ which will transform S into a tree S' isomorphic with T. (Note: for purposes of clarity, in the following we usually prefer to say (incorrectly): "...transform S into T"... .)

Number the internal nodes of a binary search tree, with n internal nodes, from 1 to n. Then we may speak of ρ_i, the clockwise rotation at node i and $\bar{\rho}_i$ as the counterclockwise rotation at i.

We now have our first result, namely:

<u>Theorem 1</u>

Let S and T be two binary search trees with $n > 1$ internal nodes. Then $0 < \text{rdist}(S,T) < 2n - 2$.

Proof: Let R =  also have n internal nodes.

We claim that $0 \leq \mathrm{rdist}(S,R) \leq n - 1$.

If R is equal to S then rdist(R,S) = 0. Otherwise the right spine of S contains $1 \leq m < n$ internal nodes, since the root is on the right spine and S is not equal to R. Consider a node i, say, on the right spine that has at least one internal node in its left subtree. Perform ρ_i to give S' with m + 1 internal nodes on its right spine. Repeating this process yields, eventually, R. Clearly at most n − 1 rotations are used. Similarily, from T we can also generate R with a sequence $\rho_{i_1}, \rho_{i_2} \ldots \rho_{i_m}$ of rotations, $0 \leq m \leq n - 1$. Hence $\bar{\rho}_{i_m} \bar{\rho}_{i_{m-1}} \ldots \bar{\rho}_{i_1}$ applied to R yields T, and this in turn yields the result. ∎

Letting J_n denote the class of binary search trees with n internal nodes, we next obtain:

Theorem 2

For all n, (J_n, rdist) forms a metric space.

Proof:

(i) $\mathrm{rdist}(T,T) = 0$, for all T in J_n.

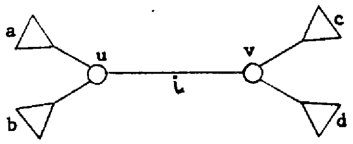
(ii) $\mathrm{rdist}(S,T) = \mathrm{rdist}(T,S)$, for all S and T in J_n.

(iii) $\mathrm{rdist}(S,T) \leq \mathrm{rdist}(S,R) + \mathrm{rdist}(R,T)$, for all R,S and T in J_n. ∎

An open problem is to determine the complexity of computing rdist(S,T).

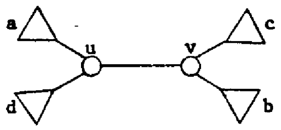
We now turn to unrooted binary trees. In [M] and [GB] the notion of a "nearest neighbor 1-step change" is introduced. This has been studied in more detail in [WS]; they call it the "nearest neighbor interchange". We will simply call this tree transformation an interchange and we define it as follows:

given

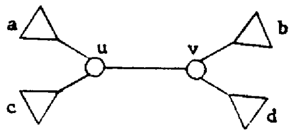


an interchange about the edge i yields:

either

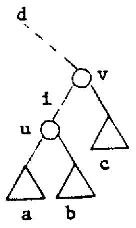


or

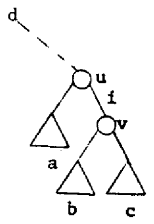


Recall that we are dealing with free trees and hence the interchange takes the set $\{a, b, c, d\}$ of subtrees and redistributes them to u and v to form a partition different from the initial one, $\{\{a, b\}, \{c, d\}\}$.

Note that a rotation is a thinly disguised interchange, since:

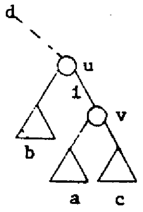


yields



that is $\{\{a, b\}, \{c, d\}\}$ yields $\{\{a, d\}, \{b, c\}\}$. This interchange is uniquely defined because the trees are ordered, and

therefore



is an invalid rotation.

We define the <u>interchange distance of S from T</u>, <u>denoted by idist(S,T)</u>, where S and T are trees with n internal nodes, as we did for rotation distance. Letting F_n denote the class of trees with n nodes, we immediately have:

<u>Theorem 3</u>

For all $n \geq 1$,

(i) For all S and T in F_n, idist(S,T) is well-defined and $0 \leq \text{idist}(S,T) \leq 2n - 2$, and

(ii) (F_n, idist) forms a metric space.

However [WS] are concerned with leaf labelled trees, that is each leaf of a given tree T has a unique label associated with it (unique with respect to T that is). It is convenient, and no loss of generality, to assume the labels are the integers 1,2,... . We use n to denote the number of leaves of a tree in this discussion, where $n > 3$.

We may once more define a distance measure between labelled trees with n leaves, let us call it the labelled interchange distance of S from T, denoted by lidist(S,T). However, note that when S and T are isomorphic, this means that they are not only structurally the same, as with idist, but also corresponding leaf nodes have the same label. We now obtain our final theorem, letting L_n denote the class of leaf labelled trees with n leaves.

Theorem 4

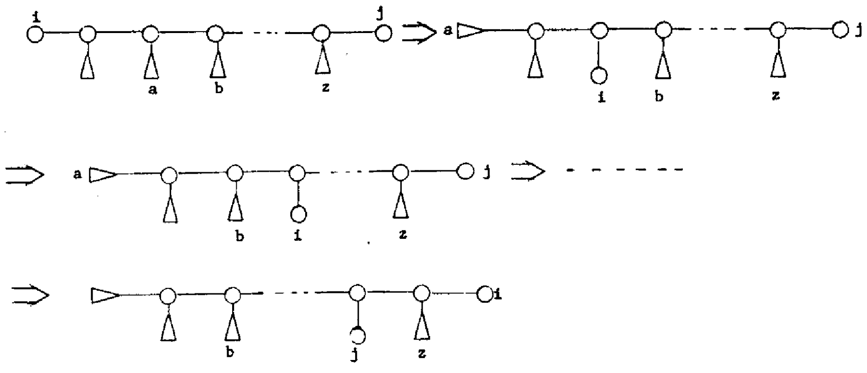
For all $n > 3$.

(i) For all S and T in L_n, lidist(S,T) is well-defined and $0 \le \text{lidist}(S,T) \le 4n - 12 + 4n[\log_2(n/3)]$, and

(ii) $(L_n$, lidist) forms a metric space.

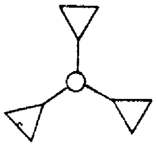
Proof

(i) lidist(S,T) is well-defined since we can transform S into R which is equal to T, if leaf labels are ignored, via Theorem 3. Then for each leaf in R with an incorrect label i, say, it is swapped with the leaf having the required label, j say:



and now j can be moved down to its position by the same technique. Clearly at most n such swaps are necessary and it should be observed that the relative ordering of the subtrees on the path connecting i and j remains undistributed by the swapping. Hence lidist(S,T) is well-defined.

To show that it is bounded above by $4n - 12 + 4n[\log_2(n/3)]$ transform S into a minimal diameter tree R, that is the longest path is minimal. Hence R has the appearance of:

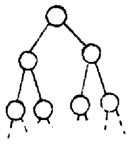


where each subtree has approximately equal height, bounded above by $[\log_2(n/3)]$. Hence to swap the values i and j at two leaves takes at most $2 \cdot (2[\log_2(n/3)])$ interchanges by the above and n swaps requires at most $4n \cdot [\log_2(n/3)]$ interchanges.

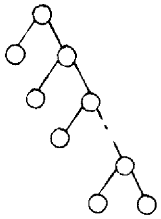
Now to obtain R, by previous arguments at most 2n - 6 interchanges are needed and similarily to obtain R from T, yielding the result.

(ii) This follows immediately ∎

As in [WS] we leave a number of problems unsolved. For the unlabelled cases we have an 0(n) interchange/rotation algorithm and this is clearly asymptotically optimal, since the tree



with n nodes requires 0(n) interchanges to give the right spine tree



However for the labelled case we have an 0(nlogn) interchange algorithm, but we have no proof of optimality, although we conjecture it to be so. Similarily the concrete bound of Theorem 4 is not known to be achievable, but we have no better one. Finally given two trees S and T what is the complexity of determining rdist(S,T), idist(S,T) or lidist(S,T)?

References

[AP] Aho, A.V., and Peterson, T.G. A minimum distance error-correcting parser for context-free languages. SIAM Journal of Computing 1 (1972), 305-312.

[D] Day, W.H.E. A new approach to constructing tree metrics. Memorial University of Newfoundland, Computer Science Technical Report No. 8001, 1980.

[H] Hirschberg, D.S. Complexity of common subsequence problems. Proceedings of the 1977 Fundamentals of Computation Theory Conference, Springer-Verlag Lecture Notes in Computer Science 56 (1977), 393-398.

[K] Knuth, D.E. The Art of Computer Programming, Volume 3: Sorting and Searching. Addison-Wesley Publishing Co., Reading, Mass., 1973.

[L] Lyon, G. Syntax-directed least-errors analysis for context-free languages: a practical approach. Communications of the ACM 17 (1974), 3-14.

[MGB] Moore, G.W., Goodman, M., and Barnabas, J. An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. Journal of Theoretical Biology 38 (1973), 423-457.

[S] Sankoff, D. Matching sequences under deletion/insertion constraints. Proceedings National Academy of Sciences USA 69, (1979), 4-6.

[WS] Waterman, M.S. and Smith, T.F. On the similarity of dendorgrams. Journal of Theoretical Biology 75 (1978), 789-800.

[WF] Wagner, R.A., and Fischer, M.J. The string-to-string correction problem. Journal of the ACM 21 (1974), 168-173.

February 16, 1982
G/USC.4/A/DW6.1
/bh